

CERTIFIED COPY OF
PRIORITY DOCUMENT

10971 U.S. PTO
09/775230
02/01/01

대한민국 특허청
KOREAN INDUSTRIAL
PROPERTY OFFICE

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto
is a true copy from the records of the Korean Industrial
Property Office.

출원번호 : 특허출원 2000년 제 4981 호
Application Number

출원년월일 : 2000년 02월 01일
Date of Application

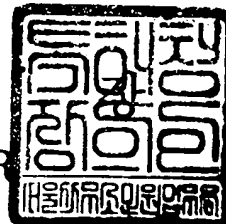
출원인 : 삼성전자 주식회사
Applicant(s)



2000 06 01 일
 년 월 일

특 허 청

COMMISSIONER



【서류명】	특허출원서		
【권리구분】	특허		
【수신처】	특허청장		
【제출일자】	2000.02.01		
【발명의 명칭】	멀티프로세싱 마이크로프로세서에 적합한 분기 예측기		
【발명의 영문명칭】	A BRANCH PREDICTOR FOR MICROPROCESSOR HAVING MULTIPLE PROCESSES		
【출원인】			
【명칭】	삼성전자 주식회사		
【출원인코드】	1-1998-104271-3		
【대리인】			
【성명】	임창현		
【대리인코드】	9-1998-000386-5		
【포괄위임등록번호】	1999-007368-2		
【대리인】			
【성명】	권혁수		
【대리인코드】	9-1999-000370-4		
【포괄위임등록번호】	1999-056971-6		
【발명자】			
【성명의 국문표기】	이회진		
【성명의 영문표기】	LEE,HOI JIN		
【주민등록번호】	710220-1167613		
【우편번호】	461-191		
【주소】	경기도 성남시 수정구 태평1동 5737번지		
【국적】	KR		
【심사청구】	청구		
【취지】	특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인 임창현 (인) 대리인 권혁수 (인)		
【수수료】			
【기본출원료】	20	면	29,000 원
【가산출원료】	6	면	6,000 원

1020000004981

2000/6/

【우선권주장료】	0	건	0	원
【심사청구료】	11	항	461,000	원
【합계】	496,000			원
【첨부서류】	1. 요약서·명세서(도면)_1통			

【요약서】**【요약】**

여기에 개시된 분기 예측기는 분기 히스토리와 분기 명령 어드레스뿐만 아니라 프로세스 ID까지 고려하여 분기 예측 참조(reference) 데이터를 저장하고 있는 분기 예측 테이블을 액세스하기 위한 인덱스를 발생한다. 따라서, 하나의 마이크로프로세서 내에서 콘텍스트 스위칭(context switching)이 자주 일어나면서 여러 가지의 프로세스들이 동시에 수행될 때에도 본 발명의 분기 예측기는 높은 분기 예측 적중률을 갖는다. 컴퓨터 시스템의 운영체제(Operating System) 입장에서 분기 예측 적중률이 향상됨에 따라 파이프라인의 정지(stall) 현상이 감소되어 프로그램 실행 시간이 단축된다.

【대표도】

도 2

【명세서】**【발명의 명칭】**

멀티프로세싱 마이크로프로세서에 적합한 분기 예측기{A BRANCH PREDICTOR FOR MICROPROCESSOR HAVING MULTIPLE PROCESSES}

【도면의 간단한 설명】

도 1은 일반적인 글로벌 히스토리 스킴을 기반으로 한 분기 예측기를 개략적으로 보여주는 블록도; 그리고

도 2는 글로벌 히스토리 스킴에 근거한 본 발명의 바람직한 실시예에 따른 분기 예측기의 구성을 보여주는 블록도이다.

도면의 주요 부분에 대한 부호의 설명

10 : 프로그램 메모리 20 : 명령어 페치 유닛

30 : 명령어 디코더 40 : 실행 유닛

50 : 분기 예측기 52 : 히스토리 레지스터

54 : 해시 로직 56 : 분기 예측 테이블

58 : 어드레스 선택 회로 60 : 분기 예측 적중/실패 검사기

【발명의 상세한 설명】**【발명의 목적】****【발명이 속하는 기술분야 및 그 분야의 종래기술】**

<9> 본 발명은 마이크로프로세서의 분기 예측기에 관한 것으로서, 더 구체적으로는 멀티-프로세싱 환경 하에서 뛰어난 적중률을 보이는 분기 예측기에 관한 것이다.

- <10> 종래의 프로세서들은 일반적으로 SISD(Single Instruction Single Data) 프로세서였다. SISD 프로세서는 단일 명령어 스트림과 단일 데이터 스트림을 받아들여 각 명령어를 순차적으로 실행한다. 그러나, 이러한 SISD 프로세서 구조는 많은 데이터를 처리해야 하는 현 시점에서는 매우 불리하다.
- <11> 프로세서의 처리량이 증가함에 따라 병렬 프로세싱 구조가 개발되었다. 그러한 병렬 프로세싱 모델 가운데 하나가 파이프라인(pipeline) 프로세싱이다. 간단한 파이프라인 프로세서에서, 파이프라인은 여러 개의 단계들을 포함한다. 이 단계들은, 예를 들면, 페치(fetch) 단계, 디코드(decode) 단계, 실행(execute) 단계이다. 상기 파이프라인 프로세서에서, 명령어들은 겹쳐진 형태로 파이프의 각 단계들을 순차적으로 지나가면서 수행된다.
- <12> 파이프라인 프로세서들의 성능은 분기(branch) 동작에 전적으로 의존한다. 분기 동작들은 프로그램 실행 도중에 명령어의 흐름 방향을 바꾸므로 파이프라인 길이를 길게 한다. 이는 분기 명령어가 페치될 때 다음 실행(페치)될 명령어의 어드레스를 즉시 알 수 없기 때문이다. 그러므로, 페치 단계는 정지(stall)되고, 분기 어드레스가 계산될 때까지 지연된다. 일반적으로 분기 어드레스는 실행 단계에서 생성된다. 따라서, 페치 단계는 파이프라인의 디코드 및 실행 단계동안 정지된다. 분기 명령어가 완전히 실행되고 나면 분기할 방향을 알 수 있고, 올바른 타겟 어드레스에서 안전하게 명령어를 페치할 수 있다. 즉, 조건부 분기 명령어(conditional branch instruction)의 조건이 참 또는 거짓인 지의 여부를 판단하고, 다음 실행될 명령어를 페치하기 위해 다음 프로그램 카운터(program counter; PC)의 값을 결정하는 시점 사이에 1 사이클(cycle) 이상의 시간차가 존재한다.

<13> 이와 같이 낭비되는 사이클을 활용하기 위해서 조건부 분기 명령어의 분기 여부를 예측하여 파이프라인을 진행시키는 '분기 예측(branch prediction)'이 채용되었다. 분기 예측은 조건부 분기 명령어의 조건이 참인지 또는 거짓인 지의 여부를 판별하여 분기될 어드레스가 계산되고 있는 동안 연속적으로 다음 명령어를 실행하기 위해 분기될 어드레스를 임의로 설정하여 진행하는 것이다. 분기 예측이 맞으면 임의로 실행한 명령어들이 옳게 실행된 것이고, 파이프라인의 정지가 발생되지 않는다. 반면, 분기 예측이 잘못되었다면 분기되어야 할 타겟으로 다시 분기해야 한다. 이 때에는 잘못 실행된 명령어들을 제거하고(flush), 올바른 분기된 명령어들을 재실행하기 위해 추가적인 지연이 발생한다. 이와 같이, 분기 예측이 실패했을 때 미리 예상하고 진행시킨 사이클이 쓸모 없게 되는 것을 분기 예측 실패 페널티(branch misprediction penalty)라 한다.

<14> 분기 예측 실패 페널티를 줄이기 위한 방법으로는 정적 분기 예측(static branch prediction)과 동적 분기 예측(dynamic branch prediction)이 있다. 정적 분기 예측은 프로그램을 실행하기 전에 컴파일러가 분기 명령어의 테이큰(taken; 분기 타겟 어드레스로 분기하는 경우)과 낫-테이큰(not-taken; 물리적으로 분기 명령어의 바로 다음 명령어로 진행하는 경우)을 판단하여 프로그램 코드를 재배열하는 방법이다. 동적 분기 예측은 프로그램의 실행 중에 과거 행적(히스토리)에 비추어 테이כן/낫-테이כן을 결정하는 방법이다. 일반적으로 동적 분기 예측 방법이 정적 분기 예측 방법에 비해 더 높은 적중률을 갖는다.

<15> 동적 분기 예측을 구현하는 구체적인 방법으로는, 각 분기 명령어의 어드레스마다 카운터를 두어 루프 명령어(예를 들면, for, do, loop 등)에서 뛰어난 적중률을 갖는 퍼-어드레스 히스토리 스킴(per-address history scheme) 및 if-then과 같은 인접 분기 명

령어에서 뛰어난 적중률을 보이는 글로벌 히스토리 스킴(global history scheme)이 있다. 하드웨어 부담 측면에서 글로벌 히스토리 스킴이 퍼-어드레스 히스토리 스킴에 비해 선호된다. 글로벌 히스토리 스킴에 근거한 분기 예측기가 Scott McFarling에 의해 1993년 6월, Western Research Laboratory의 Technical Note TN-36에 'Combining Branch Predictors' 라는 제목으로 개시되어 있다.

- <16> 그런데, 일반적인 분기 예측기는 하나의 프로세스에 대한 분기 동작만을 고려하여 설계되었다. 즉, 프로세스 ID는 고려하지 않았다. 따라서, 단일 프로세싱일 때 뛰어난 적중률을 보이는 동적 분기 예측 방법일지라도 하나의 마이크로프로세서 내에서 다수 개의 프로세스들이 동시에 수행되는 멀티-프로세싱(multi-processing) 환경 하에서는 분기 예측 적중률이 낮아질 수밖에 없다. 분기 예측 적중률이 낮아지면 분기 예측 실패 페널티가 증가하고 이는 곧 프로그램 실행 시간을 증가시키는 요인이 된다.

【발명이 이루고자 하는 기술적 과제】

- <17> 본 발명의 목적은 하나의 마이크로프로세서 내에서 다수 개의 프로세스들이 동시에 수행되는 멀티-프로세싱 환경 하에서 뛰어난 적중률을 보이는 분기 예측기를 제공하는 데 있다.

【발명의 구성 및 작용】

- <18> (구성)

- <19> 상술한 바와 같은 본 발명의 목적을 달성하기 위한 본 발명의 일 특징에 의하면, 마이크로프로세서의 멀티프로세싱 환경하에서 사용되는 분기 예측기는: 일련의 이전 분기 명령어들의 분기 히스토리를 저장하는 히스토리 레지스터, 현재 분기 명령어가 속하

는 프로세스의 ID, 상기 현재 분기 명령어의 어드레스 그리고 상기 분기 히스토리의 조합으로부터 인덱스를 생성하는 해시 로직, 분기 예측 참조 데이터들을 저장하고, 상기 해시 로직으로부터 생성된 인덱스에 저장되어 있는 분기 예측 참조 데이터를 출력하는 분기 예측 테이블, 상기 분기 예측 테이블로부터 출력되는 분기 예측 참조 데이터에 응답하여, 상기 현재 분기 명령어에 포함된 타겟 어드레스 또는 물리적으로 상기 분기 명령어의 다음에 위치하는 명령어의 어드레스 가운데 하나를 분기 예측 어드레스로 선택 출력하는 어드레스 선택 수단 그리고 상기 현재 분기 명령어의 실행 결과에 따른 실제적인 분기 어드레스와 분기 예측 어드레스를 비교하여, 상기 히스토리 레지스터에 저장된 분기 히스토리 및 상기 분기 예측 테이블에 저장된 분기 예측 참조 데이터를 갱신하는 분기 예측 결과 검사기를 포함한다.

<20> 바람직한 실시예에 있어서, 상기 분기 예측 테이블은 상기 해시 로직으로부터 생성된 인덱스에 의해 선택 가능한 복수 개의 업/다운 세츠헤이팅 카운터들로 구성된다.

<21> 바람직한 실시예에 있어서, 상기 히스토리 레지스터는 쉬프트 레지스터로 구성된다.

<22> 상기 해시 로직은, 상기 현재 분기 명령어가 속하는 프로세스의 ID, 상기 현재 분기 명령어의 어드레스 그리고 상기 분기 히스토리를 익스클루시브-오아 연산하여 상기 인덱스를 생성한다.

<23> 상기 분기 예측 결과 검사기는, 상기 현재 분기 명령어의 실행 결과에 따른 실제적인 분기 어드레스와 상기 분기 예측 어드레스가 서로 일치하는 지를 비교하고, 비교 결과에 대응하는 제어 신호를 발생하고, 상기 현재 분기 명령어의 실행 결과에 따른 실제적인 분기 어드레스와 상기 분기 예측 어드레스가 서로 일치하면 논리 '1'의 제어 신호

를, 그리고 일치하지 않으면 논리 '0'의 제어 신호를 발생한다.

- <24> 이 실시예에 있어서, 상기 어드레스 선택 수단은 상기 제어 신호가 논리 '0'일 때 상기 현재 분기 명령어의 실행 결과에 따른 실제적인 분기 어드레스를 상기 분기 예측 어드레스로 변경하여 출력한다.
- <25> 또, 상기 업/다운 세츄레이팅 카운터는 상기 제어 신호가 논리 '1'일 때 업 카운트 하고, 그리고 상기 제어 신호가 논리 '0'일 때 다운 카운트한다.
- <26> 그리고, 상기 쉬프트 레지스터는 상기 제어 신호를 삽입하여 상기 분기 예측 결과를 제 1 방향으로 쉬프트한다.
- <27> 본 발명의 목적을 달성하기 위한 본 발명의 다른 특징에 의하면, 분기 예측 참조 데이터를 저장하는 분기 예측 테이블을 참조하여 조건부 분기 명령어의 분기 어드레스를 예측하는 방법은: 상기 조건부 분기 명령어가 속하는 프로세스의 ID, 상기 조건부 분기 명령어의 어드레스 그리고 일련의 이전 분기 명령어들의 분기 히스토리의 조합으로부터 상기 분기 예측 테이블을 액세스하기 위한 인덱스를 생성하는 단계, 상기 인덱스에 응답하여 상기 분기 예측 테이블로부터 분기 예측 참조 데이터를 독출하는 단계, 상기 분기 예측 참조 데이터에 응답하여 상기 조건부 분기 명령어에 포함된 타겟 어드레스 또는 물리적으로 상기 조건부 분기 명령어 다음에 위치하는 명령어의 어드레스 가운데 하나를 분기 예측 어드레스로 선택 출력하는 단계 그리고 상기 조건부 분기 명령어의 실행 결과에 따른 실제적인 분기 어드레스에 응답하여 상기 분기 히스토리과 상기 분기 예측 테이블에 저장된 분기 예측 참조 데이터를 갱신하는 단계를 포함한다.
- <28> 본 발명의 바람직한 실시예에 따른 분기 예측 방법은, 상기 조건부 분기 명령어의

실행 결과에 따른 실제적인 분기 어드레스와 상기 분기 예측 어드레스가 서로 일치하는지의 여부를 판단하는 단계 및 상기 조건부 분기 명령어의 실행 결과에 따른 실제적인 분기 어드레스와 상기 분기 예측 어드레스가 서로 일치하지 않은 경우 상기 실제적인 분기 어드레스를 상기 분기 예측 어드레스로 변경하여 출력하는 단계를 더 포함한다.

<29> (작용)

<30> 이와 같은 구성에 의해서, 하나의 마이크로프로세서 내에서 다수 개의 프로세스들이 동시에 수행되는 멀티-프로세싱 환경에서 뛰어난 적중률을 보이는 분기 예측기가 구현된다.

<31> (실시예)

<32> 이하, 본 발명의 바람직한 실시예를 첨부된 도면 제 1 도 내지 제 3 도에 의거하여 상세히 설명한다.

<33> 도 1은 본 발명의 바람직한 실시예에 따른 마이크로프로세서 내에 구성되는 주요 구성 요소를 보여주는 블록도이다.

<34> 도 1을 참조하면, 상기 마이크로프로세서는 프로그램 메모리(10), 명령어 페치 유닛(20), 명령어 디코더(30), 실행 유닛(40), 그리고 분기 예측기(50)를 포함한다. 상기 프로그램 메모리(10)는 상기 명령어 페치 유닛(20)으로부터의 어드레스(PC)에 응답하여 매 사이클마다 명령어를 상기 명령어 페치 유닛(20)으로 제공한다. 상기 명령어 디코더(30)는 상기 명령어 페치 유닛(20)에서 페치된 명령어를 받아들여 디코드하고, 디코드된 코드를 상기 실행 유닛(40)으로 제공한다. 이 때, 디코드된 명령어가 조건부 분기 명령어이면, 상기 조건부 분기 명령어가 속하는 프로세스의 ID와 상기 조건부 분기 명령어에

포함된 타겟 어드레스(TARGET)를 분기 예측기(50)로 제공한다.

<35> 상기 프로세스 ID는, 예를 들면 4 비트 데이터(최대 16 개 프로세스들에 대응하는 ID를 나타낼 수 있음)이고, 상기 명령어 디코더(30)에서 현재 디코드된 명령어가 속하는 프로세스의 ID를 나타낸다. 예컨대, 현재 디코드된 명령어가 제 0 프로세스에 속하는 명령어이면 상기 프로세스 ID는 이진수 '0000'의 값을 가지며, 상기 현재 디코드된 명령어가 제 1 프로세스에 속하는 명령어이면 상기 프로세스 ID는 이진수 '0001'의 값을 갖는다.

<36> 상기 분기 예측기(50)는 상기 명령어 디코더(30)로부터 제공되는 프로세스 ID와 상기 명령어 페치 유닛(20)으로부터 제공되는 분기 명령어의 어드레스(PC)를 받아들여 글로벌 히스토리 스킴을 기반으로 한 분기 예측 어드레스(PREADDR)를 발생한다. 상기 분기 예측 어드레스(PREADDR)는 상기 명령어 페치 유닛(20)으로 제공된다. 상기 실행 유닛(40)은 상기 디코더(30)에서 디코드된 조건부 분기 명령어의 조건이 참 또는 거짓인지를 판별하고, 판별 결과에 따라 실제적인 분기 어드레스(NEXTADDR)를 발생한다. 상기 분기 예측기(50)는 자신이 예측한 어드레스(PREADDR)와 상기 실행 유닛(40)에서 발생된 실제적인 분기 어드레스(NEXTADDR)가 서로 일치하는지의 여부를 판별한다. 두 어드레스(PREADDR, NEXTADDR)가 서로 일치하는 경우에는 분기 예측이 적중(hit)한 것이므로 아무런 문제가 발생되지 않으나, 서로 일치하지 않는 경우에는 분기 예측이 실패(miss)한 것이므로 실제적인 분기 어드레스(NEXTADDR)로 재분기하기 위해 상기 분기 예측 어드레스(PREADDR)를 상기 실제적인 분기 어드레스(NEXTADDR)로 변경하여 출력한다. 이러한 분기 예측기의 상세 회로가 도 2에 도시되어 있다.

<37> 도 2를 참조하면, 글로벌 히스토리 스킴을 기반으로 한 본 발명의 바람직한 실시예

에 따른 분기 예측기(50)는 히스토리 레지스터(52), 해시 로직(54), 분기 예측 테이블(56), 어드레스 선택 회로(58) 그리고 분기 예측 결과 검사기(60)를 포함한다.

<38> 상기 히스토리 레지스터(52)는, 예를 들면 8 비트 레지스터이고, 최근의 분기 명령어부터 8 번째 이전 분기 명령어까지의 분기 히스토리(즉, 분기 예측 적중/실패 여부; GH)를 저장한다. 상기 히스토리 레지스터(52)의 각 비트들은 분기 예측이 적중한 경우 이진수 '1'의 값을 가지며, 분기 예측이 실패한 경우 이진수 '0'의 값을 갖는다.

<39> 상기 해시 로직(54)은 상기 명령어 디코더(30)로부터 제공되는 프로세스 ID, 상기 명령어 페치 유닛(20)으로부터 제공되는 분기 명령어의 어드레스(PC) 그리고 상기 히스토리 레지스터(52)에 저장된 분기 히스토리(GH)를 받아들여 해시(hash) 알고리즘에 따른 연산을 수행한다. 이 해시 로직(54)은 익스클루시브-오아(exclusive-OR) 로직 회로로 구성되어, 상기 프로세스 ID, 분기 명령어의 어드레스(PC) 및 분기 히스토리(GH)를 익스클루시브-오아 연산하여 출력한다. 단, 상기 프로세스 ID는 4 비트이므로, 상기 히스토리(GH)와 타겟 어드레스(TARGET)의 8 비트와 동일하도록 4 비트 데이터 '0000'을 프로세스 ID의 최상위 비트에 부가하여 연산한다. 다음 표 1은 상기 분기 히스토리(GH), 프로세스 ID, 및 분기 명령어의 어드레스(PC)를 익스클루시브-오아 연산한 결과를 예시적으로 보여주는 표이다.

<40> [표1]

<41>

분기 히스토리 (GH)	프로세스 ID (ID)	분기 명령어의 어드레스 (PC)	gshare 8/8 (종래 기술, HI)	해시 인덱스 (HI)
00000001	0000	00000000	00000001	00000001
00000000	0000	00000000	00000000	00000000
00000001	0010	00000000	00000001	00000011
00000000	0010	00000000	00000010	00000010
00000000	0000	11111111	11111111	11111111
10000000	0000	11111111	01111111	01111111
00000000	0010	11111111	11111111	11111101
10000000	0010	11111111	01111111	01111101

<42> 위 표는 글로벌 히스토리[7:0], 프로세스 ID[3:0], 분기 명령어의 어드레스[7:0]일 때, 기존 분기 예측기의 일 예인 gshare 8/8의 '(분기 명령어의 어드레스[7:0]) XOR (글로벌 히스토리[7:0])' 연산에 따른 해시 인덱스와 프로세스 ID를 이용한 본 발명의 '(분기 명령어의 어드레스[7:0]) XOR (글로벌 히스토리[7:0]) XOR (0000, 프로세스 ID[3:0])' 연산에 따른 해시 인덱스를 비교하여 보여주고 있다.

<43> 분기 예측 참조 데이터인 테이큰, 낫-테이큰을 선택하기 위해 참조(reference)로 삼을 카운터를 선택할 때 종래의 gshare는 프로세스 ID가 서로 달라졌을 때 구분을 하지 못하기 때문에 프로세스 ID가 달라지면 참조(reference)로 사용되던 카운터들을 버려야 하나, 본 발명은 이를 구분할 수 있기 때문에 이전 프로세스 ID에 관련된 참조를 버릴 필요가 없다. 즉, 분기 예측시 서로 간섭(aliasing)영향을 줄임으로써 실행 시간이 단축되고 분기 예측 히트율이 향상된다.

<44> 상기 해시 로직(54)으로부터 출력되는 패턴은 상기 분기 예측 테이블(56)의 인덱스

신호(HI)로 제공된다.

<45> 상기 분기 예측 테이블(56)은 상기 분기 명령어의 타겟 어드레스(TARGET)로 분기할 것(테이크; TAKEN)인지 또는 물리적으로 상기 분기 명령어 다음에 있는 명령어를 수행할 것(낫-테이크; NOT-TAKEN)인 지를 결정하기 위한 분기 예측 참조(reference) 데이터들을 저장한다. 이 분기 예측 테이블(56)은 각각 서로 다른 인덱스에 의해 선택되는 N개의 카운터들로 구성된다. 상기 해시 로직(54)으로부터 출력되는 패턴(HI)이 8 비트이므로 상기 분기 예측 테이블(56)은 최대 2^8 개의 카운터들로 구성된다. 상기 분기 예측 테이블(56)을 구성하는 카운터들은 각각 2 비트 업/다운 세추레이팅 카운터(saturating counter)이다. 통상의 기술자에게 잘 알려진 바와 같이, 이 2 비트 업/다운 세추레이팅 카운터는 '00', '01', '10' 그리고 '11'의 네 가지 상태를 가지며, 이 상태들은 업/다운 제어 신호에 응답하여 변화된다. 단, '00'에서 다운 제어 신호가 입력되더라도 그 상태는 '00'을 유지하며, '11'에서 업 제어 신호가 입력되더라도 '11'을 유지한다.

<46> 상기 카운터에 저장된 분기 예측 참조 데이터에 따른 테이크 또는 낫-테이크 여부가 다음 표 2에 개시되어 있다.

<47> [표 2]

<48>

분기 예측 참조(reference) 데이터	테이크/낫-테이크 예측
0 0	NOT-TAKEN
0 1	NOT-TAKEN
1 0	TAKEN
1 1	TAKEN

- <49> 상기 해시 로직(54)으로부터 출력되는 패턴(HI)에 의해 선택된 카운터에 저장되어 있는 분기 예측 참조(reference) 데이터 가운데 최상위 비트(MSB)만이 상기 어드레스 선택 로직(58)으로 제공된다.
- <50> 상기 어드레스 선택 회로(58)는 상기 분기 예측 테이블(56)로부터 제공된 1 비트의 분기 예측치가 '0'이면 물리적으로 분기 명령어 다음에 있는 명령어의 어드레스(PC+k; k는 명령어 길이, 단위 : 바이트)를 선택하고, 상기 1 비트의 분기 예측치가 '1'이면 상기 명령어 디코더(30)로부터 제공되는 분기 명령어에 포함된 타겟 어드레스(TARGET)를 선택 출력한다.
- <51> 상기 분기 예측 결과 검사기(60)는 분기 예측값(T/NT)과 상기 실행 유닛(40)의 조건 분기 판단식 결과가 서로 일치하는지의 여부를 판별해서 서로 일치하는 경우(즉, 분기 예측이 적중한 경우)에는 '1'의 값을, 그리고 두 어드레스(PREADDR, NEXTADDR)가 서로 일치하지 않을 경우(즉, 분기 예측이 실패한 경우)에는 '0'의 값을 제어 신호(H/M)로서 출력한다.
- <52> 이와 같은 방법에 의해서 발생된 '1' 또는 '0'의 값을 가지는 제어 신호(H/M)를 받아들이는 히스토리 레지스터(52)는 최하위 비트(LSB)에 상기 제어 신호(H/M)를 삽입하면서 왼쪽 쉬프트(shift left)를 수행한다. 예를 들면, 상기 히스토리 레지스터(52)에 저장된 분기 예측 결과(GH)가 이진수 '01101110'이고 현재 분기 명령어에 대한 분기 예측이 적중한 경우(즉, 상기 제어 신호가 '1'인 경우), 상기 히스토리 레지스터(120)에는 이진수 '11011101'이 저장된다.
- <53> 상기 분기 예측 테이블(56)은 상기 분기 예측 적중/실패 검사기(60)로부터 발생된 '1' 또는 '0'의 값을 가지는 제어 신호(H/M)를 받아들여 상기 해시 로직(54)으로부터 출

력되는 패턴(HI)에 의해 선택된 카운터의 데이터를 갱신한다. 즉, 상기 제어 신호(H/M)가 '1'이면 '1' 만큼 업 카운트를 수행하고, 상기 제어 신호(H/M)가 '0'이면 '1' 만큼 다운 카운트를 수행한다.

- <54> 상기 어드레스 선택 회로(58)는 상기 제어 신호(H/M)가 '1'이면 아무런 동작을 취하지 않으나, 상기 제어 신호(H/M)가 '0'이면 상기 분기 예측 테이블(56)로부터 제공된 상기 분기 예측 참조 데이터(T/NT)에 따라 분기 예측 어드레스를 변경하여 출력한다. 즉, 상기 분기 예측 참조 데이터(T/NT)가 테이블에 대응한 경우에는 상기 명령어 페치 유닛(20)으로부터 제공된 분기 명령어 다음에 위치하는 명령어의 어드레스(PC+k)를 분기 예측 어드레스(PREADDR)로 변경하여 출력한다. 반면, 상기 분기 예측 참조 데이터(T/NT)가 낫-테이블에 대응한 경우에는 상기 명령어 디코더(30)로부터 제공된 분기 명령어의 타겟 어드레스(TARGET)를 상기 분기 예측 어드레스(PREADDR)로 변경하여 출력한다.
- <55> 도 3은 본 발명의 바람직한 실시예에 따른 분기 예측기의 제어 수순을 보여주는 플로우 차트이다. 상기 분기 예측기(50)는 상기 명령어 디코더(30)로부터 프로세스 ID가 입력되고, 상기 명령어 페치 유닛(20)으로부터 분기 명령어의 어드레스(PC)가 입력되면(단계 S110) 분기 예측 동작을 개시한다.
- <56> 단계 S112에서는, 분기 예측기(50) 내의 해시 로직(54)이 상기 명령어 디코더(30)로부터 제공되는 프로세스 ID, 상기 명령어 페치 유닛(20)으로부터 제공되는 분기 명령어의 어드레스(PC) 그리고 히스토리 레지스터(52)에 저장된 분기 히스토리(GH)를 원천(source)으로 하여 해시 연산을 수행하고, 그 결과로써 분기 예측 테이블(56)의 인덱스(HI)를 생성한다. 상기 해시 로직(54)으로부터 출력되는 인덱스(HI)는 분기 예측 테이블(56)을 구성하는 복수 개의 카운터들 가운데 하나를 선택하기 위한 선택 신호이다.

- <57> 단계 S114에서는, 상기 인덱스(HI)에 의해 선택된 분기 예측 테이블(56)의 카운터로부터 분기 예측 참조 데이터(T/NT)가 독출된다.
- <58> 계속해서 단계 S116에서는, 어드레스 선택 회로(58)가 상기 분기 예측 참조 데이터(T/NT)에 응답하여, 상기 명령어 디코더로부터 제공되는 분기 명령어의 타겟 어드레스(TARGET) 또는 상기 명령어 페치 유닛(20)으로부터 제공되는 물리적으로 상기 분기 명령어 다음에 위치하는 명령어의 어드레스(PC+k) 가운데 하나를 선택해서 분기 예측 어드레스(PREADDR)를 발생한다. 한편, 명령어 실행 유닛(40)은 상기 조건부 분기 명령어를 실행하고 나서 실제적인 분기 어드레스(NEXTADDR)를 생성한다.
- <59> 단계 S118에서, 분기 예측 결과 검사기(60)는 상기 명령어 실행 유닛(40)으로부터 출력되는 실제적인 분기 어드레스(NEXTADDR)와 상기 어드레스 선택 회로(58)로부터 출력된 분기 예측 어드레스(PREADDR)가 서로 일치하는지의 여부를 판별함으로써 분기 예측이 적중(hit) 또는 실패(miss)하였는지의 여부를 판별한다.
- <60> 분기 예측이 적중(hit)한 경우 그 제어는 단계 S120으로 진행하여 분기 예측 테이블(58)에 저장된 분기 예측 참조 데이터를 '1' 만큼 증가시키고, 상기 히스토리 레지스터(52)에 저장된 분기 예측 결과의 최하위 비트(LSB)에 '1'을 삽입하면서 왼쪽 쉬프트를 수행한다(단계 S122).
- <61> 분기 예측이 실패(miss)한 경우, 그 제어는 단계 S130으로 진행하여 분기 예측 어드레스를 변경하여 출력한다. 낫-테이큰으로 분기 예측한 경우에는 상기 명령어 디코더(30)로부터 제공된 분기 명령어의 타겟 어드레스(TARGET)를, 그리고 테이큰으로 분기 예측한 경우에는 상기 메모리 페치 유닛(20)으로부터 제공된 분기 명령어 다음에 위치하는 명령어의 어드레스(PC+k)를 분기 예측 어드레스(PREADDR)로 변경하여 출력한다

. 상기 변경된 분기 예측 어드레스(PREADDR)는 상기 명령어 페치 유닛(20)으로 제공된다. 계속해서, 분기 예측 테이블(58)에 저장된 분기 예측 참조 데이터를 '1' 만큼 감소시키고(단계 S132), 상기 히스토리 레지스터(52)에 저장된 분기 예측 결과의 최하위 비트(LSB)에 '0'을 삽입하면서 왼쪽 쉬프트를 수행한다(단계 S134).

<62> 분기 예측 적중률을 높이기 위한 방법 가운데 하나는 분기 예측 참조 데이터를 저장하는 분기 예측 테이블(56)을 정확하게 액세스하는 것이다. 즉, 상기 분기 예측 테이블(56)을 구성하는 2^8 개의 카운터들 가운데 어느 것이 현재 디코드된 분기 명령어의 테이블/나트-테이블 정보를 저장하고 있는 지를 정확하게 선택해야 한다.

<63> 여러개의 프로세스가 한 컴퓨터 시스템에서 스케줄 알고리즘(schedule algorithm)에 따라 콘텍스트 스위칭(context switching)하면서 실행되는 경우, 종래의 분기 예측기가 학습(learning) 시킨 분기 예측 테이블은 쓸모 없게 되므로 모두 무효(invalid)로 되고 다시 분기 예측 테이블을 학습(learning)시켜야 한다. 여기서 학습(learning)의 의미는 현재 수행하고 있는 프로세스에 따라 분기 예측률이 향상되도록 분기 예측 테이블의 분기 예측 참조 데이터를 갱신 및 추가하는 과정을 일컫는다. 따라서, 분기 예측기가 일정 분기 예측률을 낼 수 있기까지 소요되는 학습 시간은 반드시 필요하므로 여러 프로세스를 함께 실행시키는 OS 입장에서 봤을 때 분기 예측률이 높을 수 없다. 본 발명은 콘텍스트 스위칭이 있더라도 프로세스 ID로 분기 예측 테이블의 분기 예측 참조 데이터를 선별 사용할 수 있기 때문에 분기 예측 테이블을 모두 무효(invalid) 시킬 필요가 없다. 즉, 콘텍스트 스위칭시 마다 수반되던 분기 예측 학습 시간(branch predictor learning time)을 없앨 수 있다. 따라서 컴퓨터 시스템의 OS 입장에서 봤을 때 분기 예측률이 기존 분기 예측기보다 월등하다.

<64> 이와 같이 본 발명의 분기 예측기(50)는 분기 예측 결과(GH)와 분기 명령어의 어드레스(PC)뿐만 아니라 프로세스 ID까지 고려하여 분기 예측 테이블(56)을 액세스하기 위한 인덱스(HI)를 발생한다. 따라서, 하나의 마이크로프로세서 내에서 여러 가지의 프로세스들이 동시에 수행될 때에도 본 발명의 분기 예측기(50)는 높은 분기 예측 적중률을 갖는다.

<65> 이 실시예에서는 글로벌 히스토리 스킴에 기반을 둔 분기 예측기를 도시하고 설명하였으나 이는 예를 들어 설명한 것에 불과하며 어떤 다이내믹 분기 예측기(예를 들면, gshare, combined, bi-mode, per-address 등)에서도 적용 가능하다. 또한, 본 발명의 기술적 사상을 벗어나지 않는 범위 내에서 다양한 변화 및 변경이 가능함은 물론이다.

【발명의 효과】

<66> 상술한 바와 같은 본 발명의 분기 예측기는 히스토리와 타겟 어드레스뿐만 아니라 프로세스 ID까지 고려하여 분기 예측 테이블을 액세스하기 위한 인덱스를 발생한다. 따라서, 하나의 마이크로프로세서 내에서 여러 가지의 프로세스들이 동시에 수행될 때에도 본 발명의 분기 예측기는 높은 분기 예측 적중률을 갖는다. 분기 예측 적중률이 향상됨에 따라 파이프라인의 정지(stall) 현상이 감소되어 프로그램 실행 시간이 단축된다.

【특허청구범위】**【청구항 1】**

컴퓨터의 멀티프로세싱 환경에서 사용되는 분기 예측기에 있어서:

일련의 이전 분기 명령어들의 분기 히스토리를 저장하는 히스토리 레지스터와;

현재 분기 명령어가 속하는 프로세스의 ID, 상기 현재 분기 명령어의 어드레스 그리고 상기 분기 히스토리의 조합으로부터 인덱스를 생성하는 해시 로직과;

분기 예측 참조 데이터들을 저장하고, 상기 해시 로직에서 생성된 인덱스에 대응하는 분기 예측 참조 데이터를 출력하는 분기 예측 테이블과;

상기 분기 예측 테이블로부터 출력되는 분기 예측 참조 데이터에 응답하여, 상기 현재 분기 명령어에 포함된 타겟 어드레스 또는 물리적으로 상기 현재 분기 명령어의 다음에 위치하는 명령어의 어드레스 가운데 하나를 분기 예측 어드레스로 선택 출력하는 어드레스 선택 수단; 그리고

상기 현재 분기 명령어의 실행 결과에 따른 실제적인 분기 어드레스와 상기 분기 예측 어드레스에 응답하여, 상기 히스토리 레지스터에 저장된 분기 히스토리 및 상기 분기 예측 테이블에 저장된 분기 예측 참조 데이터를 갱신하는 분기 예측 결과 검사기를 포함하는 것을 특징으로 하는 분기 예측기.

【청구항 2】

제 1 항에 있어서,

상기 분기 예측 테이블은 상기 해시 로직으로부터 생성된 인덱스에 의해 선택 가능

한 복수 개의 업/다운 세췌레이팅 카운터들로 구성되는 것을 특징으로 하는 분기 예측기.

【청구항 3】

제 1 항에 있어서,

상기 히스토리 레지스터는 쉬프트 레지스터로 구성되는 것을 특징으로 하는 분기 예측기.

【청구항 4】

제 1 항에 있어서,

상기 해시 로직은, 상기 현재 분기 명령어가 속하는 프로세스의 ID, 상기 현재 분기 명령어의 어드레스 그리고 상기 분기 히스토리를 익스클루시브-오아 연산하여 상기 인덱스를 생성하는 것을 특징으로 하는 분기 예측기.

【청구항 5】

제 1 항에 있어서,

상기 분기 예측 결과 검사기는,

상기 현재 분기 명령어의 실행 결과에 따른 실제적인 분기 어드레스와 상기 분기 예측 어드레스가 서로 일치하는 지를 비교하고, 비교 결과에 대응하는 제어 신호를 발생 하는 비교기를 포함하는 것을 특징으로 하는 분기 예측기.

【청구항 6】

제 5 항에 있어서,

상기 비교기는 상기 현재 분기 명령어의 실행 결과에 따른 실제적인 분기 어

드레스와 상기 분기 예측 어드레스가 서로 일치하면 논리 '1'의 제어 신호를, 그리고 일치하지 않으면 논리 '0'의 제어 신호를 발생하는 것을 특징으로 하는 분기 예측기.

【청구항 7】

제 6 항에 있어서,

상기 어드레스 선택 수단은,

상기 제어 신호가 논리 '0'일 때 상기 현재 분기 명령어의 실행 결과에 따른 실제적인 분기 어드레스를 상기 분기 예측 어드레스로 변경하여 출력하는 것을 특징으로 하는 분기 예측기.

【청구항 8】

제 6 항에 있어서,

상기 카운터는, 상기 제어 신호가 논리 '1'일 때 업 카운트하고, 그리고 상기 제어 신호가 논리 '0'일 때 다운 카운트하는 것을 특징으로 하는 분기 예측기.

【청구항 9】

제 6 항에 있어서,

상기 쉬프트 레지스터는 상기 제어 신호를 삽입하여 상기 분기 예측 결과를 제 1 방향으로 쉬프트하는 것을 특징으로 하는 분기 예측기.

【청구항 10】

분기 예측 참조 데이터를 저장하는 분기 예측 테이블을 참조하여 조건부 분기 명령어의 분기 어드레스를 예측하는 방법에 있어서:

상기 조건부 분기 명령어가 속하는 프로세스의 ID, 상기 조건부 분기 명령어의 어

드레스 그리고 일련의 이전 분기 명령어들의 분기 히스토리의 조합으로부터 상기 분기 예측 테이블을 액세스하기 위한 인덱스를 생성하는 단계와;

상기 인덱스에 응답하여 상기 분기 예측 테이블로부터 분기 예측 참조 데이터를 독출하는 단계와;

상기 분기 예측 참조 데이터에 응답하여 상기 조건부 분기 명령어에 포함된 타겟 어드레스 또는 물리적으로 상기 조건부 분기 명령어 다음에 위치하는 명령어의 어드레스 가운데 하나를 선택적으로 출력하는 단계; 그리고

상기 조건부 분기 명령어의 실행 결과에 따른 실제적인 분기 어드레스에 응답하여 상기 분기 히스토리와 상기 분기 예측 테이블에 저장된 분기 예측 참조 데이터를 갱신하는 단계를 포함하는 것을 특징으로 하는 조건부 분기 명령어의 분기 예측 방법.

【청구항 11】

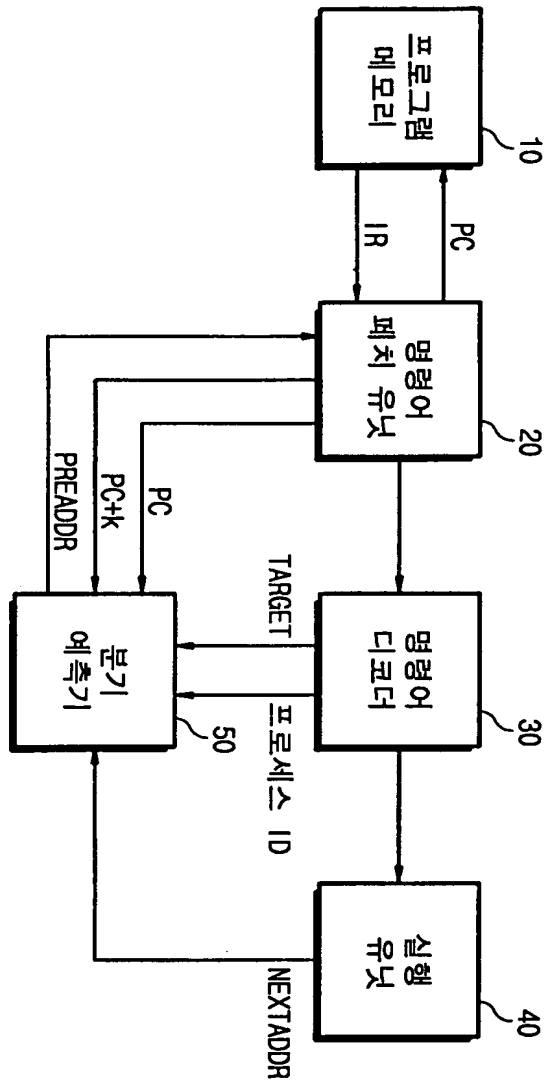
제 10 항에 있어서,

상기 조건부 분기 명령어의 실행 결과에 따른 실제적인 분기 어드레스와 상기 분기 예측 어드레스가 서로 일치하는지의 여부를 판단하는 단계; 및

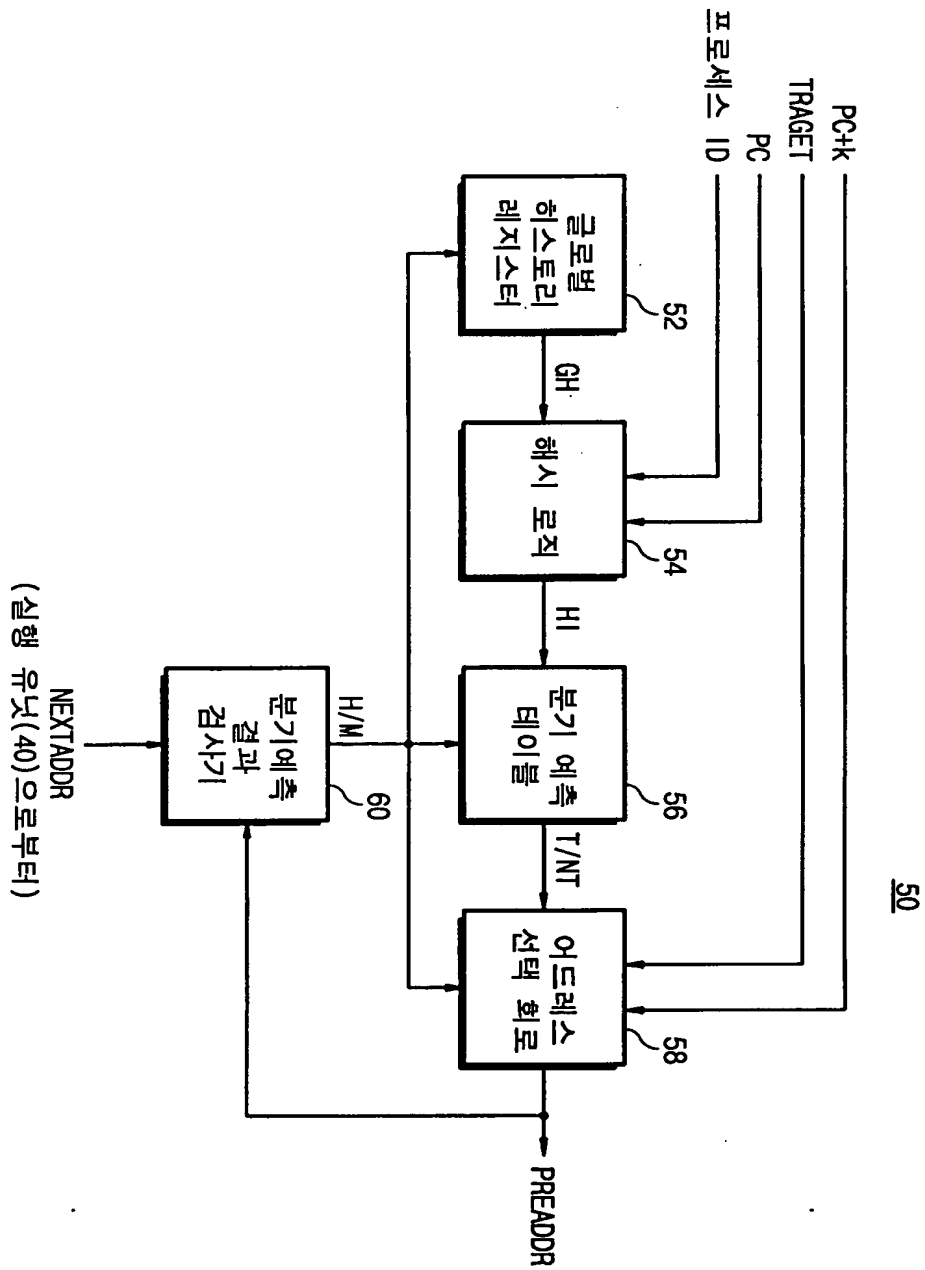
상기 조건부 분기 명령어의 실행 결과에 따른 실제적인 분기 어드레스와 상기 분기 예측 어드레스가 서로 일치하지 않은 경우 상기 실제적인 분기 어드레스를 상기 분기 예측 어드레스로 변경하여 출력하는 단계를 더 포함하는 것을 특징으로 하는 조건부 분기 명령어의 분기 예측 방법.

【도면】

【도 1】



【도 2】



【도 3】

